**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1. Which of the following is true about a trivial functional dependency?
   1. **It is always satisfied**
   2. It is never satisfied
   3. It is satisfied in some cases
   4. It is satisfied only for certain attribute combinations
2. Which of the following is not a property of a relation in a relational model?
   1. Atomicity
   2. Consistency
   3. Integrity
   4. **Duplicity**
3. Which technique is used to improve the efficiency of query processing in a DBMS?
   1. **Indexing**
   2. Encryption
   3. Replication
   4. Partitioning
4. Which database security measure provides a way to track and record all changes made to the database?
   1. Encryption
   2. **Auditing**
   3. Backup and recovery
   4. Firewall
5. Which SQL command is used to update data in a database table?
   1. INSERT INTO
   2. MODIFY
   3. ALTER TABLE
   4. **UPDATE**
6. Which of the following is a technique used to prevent unauthorized access to data during transmission?
   1. Intrusion detection system
   2. **Virtual private network (VPN)**
   3. Public key infrastructure (PKI)
   4. Access control list (ACL)
7. What happens when a cursor is opened in DBMS?
   1. **The cursor is created and memory is allocated.**
   2. The cursor is closed and the result set is discarded.
   3. The cursor is positioned at the first row of the result set.
   4. The cursor is deleted from the database.
8. When should cursors be used in DBMS?
   1. **When retrieving a large amount of data at once**
   2. When executing a single SQL statement
   3. When working with a static result set
   4. When optimizing query performance
9. Which control structure is used to exit the loop and skip the remaining iterations?
   1. **BREAK**
   2. CONTINUE
   3. EXIT
   4. RETURN
10. Which of the following is not a key constraint in the process of normalization?
    1. Functional dependency preservation
    2. **Atomicity**
    3. Lossless-join property
    4. Multi-valued dependency preservation

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

1. When the values in one or more attributes being used as a foreign key must exist in another set of one or more attributes in another table, we have created a(n):
   1. Transitive Dependency
   2. Insertion Anomaly
   3. **Referential Integrity Constraint**
   4. Normal Form
2. Which of the following are TCL commands?
   1. **COMMIT and ROLLBACK**
   2. UPDATE and TRUNCATE
   3. SELECT and INSERT
   4. GRANT and REVOKE
3. The SQL statement:

SELECT ROUND (65.726, -1) FROM DUAL;

Prints:

1. is illegal
2. garbage
3. 726
4. **70**
5. Which of the following are the synonyms for Column and ROW of a table?

Row = [Tuple, Record]

Column = [Field, Attribute]

Row = [Tuple, Attribute]

Columns = [Field, Record]

1. **1 and 2**
2. 3 and 4
3. Only 1
4. Only 2
5. Evaluate the SQL statement:

SELECT ROUND (TRUNCATE (MOD (1600, 10), -1), 2) FROM dual;

What will be displayed?

* 1. **0**
  2. 1
  3. 00
  4. An error statement

**SECTION-C(Coding Question) (4x5 marks=20 marks)**

1. Create table customers with columns like 'customer\_id', 'first\_name', 'last\_name', 'eamil' and ' phone'. Add multiple records in it. Create table products with attributes like: product\_id, product\_name, price, stock\_quantity. Add multiple records in it. Create orders table with foreign key customer\_id from customers table, add one order record in orders table.

Solution:

**CREATE TABLE customers (**

**customer\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**email VARCHAR(100),**

**phone VARCHAR(20)**

**);**

**-- insert record**

**INSERT INTO customers (customer\_id, first\_name, last\_name, email, phone)**

**VALUES (1, 'John', 'Doe', 'john@example.com', '555-123-4567'),**

**(2, 'Siya', 'Shashtri', 'siya@example.com', '543-163-7864'),**

**(3, 'Rehan', 'Mishra', 'rehan@example.com', '643-183-9462');**

**CREATE TABLE products (**

**product\_id INT PRIMARY KEY,**

**product\_name VARCHAR(100),**

**price DECIMAL(10, 2),**

**stock\_quantity INT**

**);**

**-- insert multiple records**

**INSERT INTO products (product\_id, product\_name, price, stock\_quantity)**

**VALUES**

**(1, 'Widget A', 19.99, 100),**

**(2, 'Widget B', 24.99, 75),**

**(3, 'Widget C', 29.99, 50);**

**-- create orders table with foreign key customer\_id from customers table**

**CREATE TABLE orders (**

**order\_id INT PRIMARY KEY,**

**customer\_id INT,**

**order\_date DATE,**

**total\_amount DECIMAL(10, 2),**

**FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)**

**);**

**-- insert record in order table**

**INSERT INTO orders (order\_id, customer\_id, order\_date, total\_amount)**

**VALUES (1, 1, '2023-07-28', 43.98);**

**select \* from orders;**

1. Create table customers with columns like 'customer\_id', 'first\_name', 'last\_name', 'email' and ' phone'. Insert one record in customers table and display all the records in it.Delete specific customer's record.

Solution:

**CREATE TABLE customers (**

**customer\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**email VARCHAR(100),**

**phone VARCHAR(20)**

**);**

**-- insert record**

**INSERT INTO customers (customer\_id, first\_name, last\_name, email, phone)**

**VALUES (1, 'John', 'Doe', 'john@example.com', '555-123-4567');**

**-- display records**

**select \* from customers;**

**-- delete record**

**DELETE FROM customers**

**WHERE customer\_id = 1;**

**-- display records**

**select \* from customers;**

1. Create table Books with attributes book\_id, title, category\_id, price, published\_date

'Books' table is in 1NF already. You have to remove the partial dependency by separating book price

from the Books table to a new table.

Solution:

**-- Create Books table**

**CREATE TABLE Books (**

**book\_id INT PRIMARY KEY,**

**title VARCHAR(100),**

**category\_id INT,**

**price DECIMAL(10, 2),**

**published\_date DATE**

**);**

**-- 2 NF**

**-- Create a new table for BookPrices**

**CREATE TABLE BookPrices (**

**book\_id INT PRIMARY KEY,**

**price DECIMAL(10, 2)**

**);**

**-- Insert records into BookPrices table**

**INSERT INTO BookPrices (book\_id, price)**

**VALUES (1001, 19.99),**

**(1002, 14.99),**

**(1003, 12.50),**

**(1004, 24.99),**

**(1005, 9.99);**

**-- Remove price from the Books table**

**ALTER TABLE Books**

**DROP COLUMN price;**

1. Write a PL/SQL program to arrange the number of two variable in such a way that the

small number will store in num\_small variable and large number will store in num\_large variable.

Solution:

**DECLARE**

**num\_small NUMBER := 8;**

**num\_large NUMBER := 5;**

**num\_temp NUMBER;**

**BEGIN**

**IF num\_small > num\_large THEN**

**num\_temp := num\_small;**

**num\_small := num\_large;**

**num\_large := num\_temp;**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE ('num\_small = '||num\_small);**

**DBMS\_OUTPUT.PUT\_LINE ('num\_large = '||num\_large);**

**END;**

**/**